LAB0 Assignment – ESE516-SPRING 2019

DUE DATE: JANUARY 27RD 2019 before 11:59pm EST. To be submitted on CANVAS INDIVIDUALLY

Remember: Submit your content through CANVAS. **UPLOAD a ZIP FILE with the following contents:**

- **Zip File in the format NAME_PENNKEY_LAB0.zip (Example: EduardoGarcia_Edgarc_LAB0.zip)**
  - **Excel sheet for question 1**
  - **Atmel Studio Project for question 2**

LAB0 Assignment Description

1.) BOM and Power Budget – First Pass [30 points]

- Now that you have a potential BOM for your project, it is now time to start thinking about powering them. Add the following columns to your BOM and submit the updated BOM:

- **Power Source Column:** For each component, check what power source you need to use to power it. Assume we are using a LiPo with an operating voltage from 3.4V to 4.2V. Check if you need an LDO, a Boost, a Buck, or a Buck/Boost converter, and to what voltage. **Assume we are powering the device with a one cell LiPo Battery. What voltage range does a LiPo battery usually go from?**

- **Current Consumption (One column per expected operation "mode" of your device. You must add at least one column!"):** This column tries to estimate the current that each component will consume (you can find this on the datasheet). If a certain component is not turned on in said mode, put the sleep current/idle current of the device. The example below has a column for an operation of "take picture" and another one for the operation of "water plant". We will not run the camera and the motor at the same time on the below example.
- **Current Consumption Sleep Mode:** IoT devices generally have a sleep mode, or a low power mode, to save battery when we are not functioning. Estimate the current consumption of your device when it enters its sleep mode. **If the device does not have an specific sleep mode/standby mode / low power mode in its datasheet, put the current that it consumes when it is left normally on!**
  - We will see techniques to deal with circuits that don't have a specific sleep mode.

Component	Manufacturer part number (MPN)	Manufacturer	Digikey PN	Voltage (Min)	Voltage(Max)	Power Source	Current Consumption Take Picture Operation [mA]	Current Consumption Water Plants Operation [mA]	Current Consumption - Low Power Mode (Sleep) [mA]
MCU and WiFI	ATSAMW25H18-	Microchip	ATSAMW25H18	2.7V	4.3V	3.3V Buck/Boost	4.92 mA (SAMD21) + 230mA (winc1500)	4.92 mA (SAMD21) + 230mA (winc1500)	0.5uA(WINC1500) + 12.8
Soil Moisture Sensor	SEN-13637	Sparkfun	1568-1670-ND	2.7V	4.3V (MCU	3.3V Buck/Boost	0.31mA	0.31mA	50nA
Motor Driver	DRV8833PWP	TI	296-29434-2-N	2.7v	10.8V	6V Boost and 3.3V Buck Boost	2.5uA	3	2.5uA
Water Pump	FIT0563	DRFROBOT	1738-1398-ND	6V	12V	6V Boost	0	1000mA	0
Camera	Waveshare OV	WaveShare	Available from A	2.7V	3.3V	3.3V Buck/Boost	27mA	20uA	20uA

2.) Programming Warm-Up: Simple Cypher [70 points]

Building on the last example from LAB0, make a program where the user types an ASCII character, and the system returns the ASCII numeric character code (and a next line, so we can easily read the response).

Here are some examples:

*If the user types ‘0’ =➔ Your program returns ‘48/r/n’

*If the user types ‘a’ =➔ Your program returns ‘97/r/n’

*If the user types ‘A’ =➔ Your program returns ‘65/r/n’

Use the following table as reference. Here are some tips:

- What do you have to change from the last example of LAB0 to echo a string at every character received, instead of every 4?
- How is a character represented in a byte? Is it the ASCII decimal value?
- Use the C library for strings to help you convert the character into a decimal! Search for “*snprintf*” online and see how it works.
    - Question: Why is *snprintf*, which asks you the maximum bytes allowed to be written in the given pointer, useful for embedded devices? Just think about it, no need to submit the answer to this question!

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com